

REMARKS

In the Office Action ("OA"), the Examiner rejected claims 1, 3-6, 9-10, 12-15, and 18-22 under 35 U.S.C. § 103(a) as being unpatentable over McLennan, Michael J., "Object-oriented Programming with [incr Tcl] Building Mega-Widgets with [incr TK]" (*McLennan*) in view of U.S. Patent No. 6,047,284 to Owens et al. (*Owens*); rejected claims 2 and 11 as being unpatentable over *McLennan* in view of *Owens*, and further in view of U.S. Patent No. 5,943,496 to Li et al. (*Li*); rejected claims 7-8 and 16-17 as being unpatentable over *McLennan* in view of *Owens*, and further in view of Hostetter et al., "Curl: A Gentle Slope Language for the Web" (*Hostetter*).

In the application, claims 1-22 are currently pending. In view of the foregoing amendments and the following remarks, Applicants respectfully traverse the Examiner's rejections of the claims under 35 U.S.C. § 103(a).

To establish a prima facie case of obviousness, three basic criteria must be met. First, the prior art reference as modified must teach or suggest all the claim elements. (See M.P.E.P. § 2143.03 (8th ed. 2001)). Second, there must be some suggestion or motivation, either in the reference or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or combine the reference teachings. (See M.P.E.P. § 2143 (8th ed. 2001)). Third a reasonable expectation of success must exist. Moreover, each of these requirement must "be found in the prior art, and not be based on applicant's disclosure." (M.P.E.P. § 2143.03 (8th ed. 2001)).

Claim 1 provides for:

A method of processing data comprising:

defining an object with an option data structure which supports references to option values without preallocation of memory space for the full option values; and

notifying objects of a change in an option value through a change handler identified by an option binding, the option binding being located by first searching a mapping data structure for a previously computed mapping to the option binding and, if no mapping was previously computed, by then computing the mapping to the option binding and storing the mapping in the mapping data structure.

Applicants respectfully submit that neither *McLennan* nor *Owens* disclose or suggest this claimed combination of steps. For example, the references do not disclose or suggest at least “notifying objects of a change in an option value through a change handler identified by an option binding, the option binding being located by first searching a mapping data structure for a previously computed mapping to the option binding and, if no mapping was previously computed, by then computing the mapping to the option binding and storing the mapping in the mapping data structure,” as recited in claim 1.

With respect to claim 1, the Examiner alleged that *McLennan* discloses “notifying objects of a change in an option value through a change handler identified by an option binding, the option binding being located by first searching a mapping data structure for a previously computed mapping to the option binding and, if no mapping was previously computed, by then computing the mapping to the option binding and storing the mapping in the mapping data structure” (OA pg. 3-4). Applicants respectfully disagree.

McLennan discloses methods for building mega-widgets with [incr Tk] (*McLennan* title). The methods describe using normal Tk widgets as components to create brand new widgets, or mega-widgets, in [incr Tk] (*McLennan* pg. 71, lines 8-12).

When a mega-widget is constructed, the constructor for the least-specific class itk::Archetype is called first (*McLennan* pg. 76, lines 13-15). When called, the constructor initializes some internal variables that keep track of component widgets and their configuration options (*Id.*). As each component is created, the component's options are merged into a master list of options for the mega-widget (*McLennan* pg. 77, lines 1-2; figure 2-5). A call to the itk_component adds the component and merges the component's configuration options into the master list of options (*McLennan* pg. 78, lines 29-31). A series of commands, such as "keep", at the end of the itk_component call controls how the component's configuration options are merged into the master list (*McLennan* pg. 78, lines 35-36; Figures 2-6 to 2-7). When one sets an option on the master option list, it affects all of the internal components that are tied to that option (*McLennan* pg. 80, lines 2-4). The rest of the component options will be ignored by default and not merged to the master options list, which is preferable for certain component options such as -text or -component as they are different for each components in the mega-widget (*McLennan* pg. 81, lines 10-14).

However, neither these teachings, nor any other teachings of *McLennan*, can constitute "notifying objects of a change in an option value through a change handler identified by an option binding, the option binding being located by first searching a mapping data structure for a previously computed mapping to the option binding and, if no mapping was previously computed, by then computing the mapping to the option binding and storing the mapping in the mapping data structure," as recited in claim 1. For example, *McLennan* discloses that one must include a "keep" statement, when adding a component, for each of the component one wishes to access on the

master options list (*McLennan* pg. 81, lines 8-9; Figures 2-6 to 2-8). Component options not previously merged onto the master option list will be ignored and not merged to the master options list. When a change is made to a component option not merged onto the master option list, that component option is not merged into and stored on the master option list. Therefore, *McLennan* does not constitute a teaching of “notifying objects of a change in an option value through a change handler identified by an option binding, the option binding being located by first searching a mapping data structure for a previously computed mapping to the option binding and, if no mapping was previously computed, by then computing the mapping to the option binding and storing the mapping in the mapping data structure,” as recited in claim 1.

Applicants respectfully submit that *Owens* is not sufficient to overcome the deficiencies of *McLennan*. Specifically, *Owens*, alone or in combination with *McLennan*, does not teach or suggest at least “notifying objects of a change in an option value through a change handler identified by an option binding, the option binding being located by first searching a mapping data structure for a previously computed mapping to the option binding and, if no mapping was previously computed, by then computing the mapping to the option binding and storing the mapping in the mapping data structure,” as recited in claim 1. For at least the foregoing reasons, Applicants submit that claim 1 is allowable over *McLennan* in view of *Owens*.

Claims 19, 20, and 22 and amended claim 10 are independent claims that, although of different scope, recite language similar to that which distinguishes claim 1 from *McLennan* in view of *Owens*. Accordingly, Applicants submit that claims 19, 20, and 22 and amended claim 10 are allowable over *McLennan* in view of *Owens*, for at

least the reasons given with respect to claim 1. Dependent claims 3-6, 9, 12-15, 18, and 21 are allowable not only for the reasons stated above with regard to their respective allowable base claims, but also for their own additional features that distinguish them from *McLennan* in view of *Owens*.

The Examiner rejected claims 2 and 11 under 35 U.S.C. 103(a) as unpatentable over *McLennan* in view of *Owens*, and further in view of *Li*. Applicants respectfully traverse this rejection as well.

Claims 2 and 11 are dependent claims depending on independent claims 1 and 10, respectively. Applicants respectfully submit that *Li* is not sufficient to overcome the deficiencies of *McLennan* in view of *Owens*. Specifically, *Li*, alone or in combination with *McLennan* in view of *Owens*, does not teach or suggest at least “notifying objects of a change in an option value through a change handler identified by an option binding, the option binding being located by first searching a mapping data structure for a previously computed mapping to the option binding and, if no mapping was previously computed, by then computing the mapping to the option binding and storing the mapping in the mapping data structure”, as recited in claim 1. Therefore, claims 2 and 11 are allowable over *McLennan* in view of *Owens*, and further in view of *Li*, for reasons similar to those stated above with respect to independent claims 1 and 10. Moreover, Applicants submit that claims 2 and 11 are allowable not only for the reasons stated above with regard to independent claims 1 and 10, but also for their own additional features that distinguish claims 2 and 11 from *McLennan*, *Owens*, and *Li*.

The Examiner rejected claims 7-8 and 16-17 under 35 U.S.C. 103(a) as unpatentable over *McLennan* in view of *Owens*, and further in view of *Hostetter*.

Applicants respectfully traverse this rejection as well.

Claims 7-8 and 16-17 are dependent claims depending on independent claims 1 and 10, respectively. Applicants respectfully submit that *Hostetter* is not sufficient to overcome the deficiencies of *McLennan* in view of *Owens*. Specifically, *Hostetter*, alone or in combination with *McLennan* in view of *Owens*, does not teach or suggest at least “notifying objects of a change in an option value through a change handler identified by an option binding, the option binding being located by first searching a mapping data structure for a previously computed mapping to the option binding and, if no mapping was previously computed, by then computing the mapping to the option binding and storing the mapping in the mapping data structure”, as recited in claim 1. Therefore, claims 7-8 and 16-17 are allowable over *McLennan* in view of *Owens*, and further in view of *Hostetter*, for reasons similar to those stated above with respect to independent claims 1 and 10. Moreover, Applicants submit that claims 7-8 and 16-17 are allowable not only for the reasons stated above with regard to independent claims 1 and 10, but also for their own additional features that distinguish claims 7-8 and 16-17 from *McLennan*, *Owens*, and *Hostetter*.

In view of the foregoing amendments and remarks, Applicant respectfully requests reconsideration and reexamination of this application and the timely allowance of the pending claims.

Please grant any extensions of time required to enter this response and charge any additional required fees to our deposit account 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: **April 19, 2005**

By: 

Joshua C. Liu

Reg. No. 55,391